UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/593,375	02/26/2007	Yousuke Sakao	20266	1278

23389 7590 08/17/2011

SCULLY SCOTT MURPHY & PRESSER, PC
400 GARDEN CITY PLAZA
SUITE 300
GARDEN CITY, NY 11530

EXAMINER	
COLUCCI, MICHAEL C	

ART UNIT	PAPER NUMBER
2626	

MAIL DATE	DELIVERY MODE
08/17/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/593,375	SAKAO ET AL.
	Examiner	Art Unit
	MICHAEL COLUCCI	2626

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 June 2011*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4,5,7-10,12,13,15,16 and 18-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4,5,7-10,12,13,15,16 and 18-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)	
1) ☐ Notice of References Cited (PTO-892)	4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) ☐ Notice of Informal Patent Application
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.	6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments filed 06/21/2011 have been fully considered but they are not persuasive.

Argument (page 4 ¶ 3):

- "In other words, Kumai neither discloses nor teaches "means for generating a sentence structure representing a dependency among words from an input document", "means for generating a similar structure of patterns having a similar meaning of a partial structure of the sentence structure by performing predetermined conversion operation", and "means for determining the patterns having the similar meaning as the identical pattern and detecting the patterns." In addition, Kumai neither discloses nor teaches "the parallel modification" carrying out the change in connection of the branches so that connecting relationships of the branches in the sentence structure are equal to each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure, as conceded by the Examiner"

Response to argument:

Examiner disagrees and maintains the use of Kumai. The claim language is open to the broadest reasonable interpretation, and Examiner believes that Kumai reads upon the claims, for instance, Kumai teaches a resemblance sentence retrieving tool for retrieving a sentence which is resembled to a

designated sentence. The resemblance sentence retrieving tool can acquire, for example, a list of sentences which are resembled to one designated sentence from a document obtained by an entire sentence retrieving tool. For instance, the resemblance sentence retrieving tool owns such a function that differences contained in transcription such as "PC" and "personal computer" are absorbed by employing a synonym dictionary, and while a featured word contained in a designated sentence is extracted, documents having similar featured words to the extracted featured word are provided in a top priority (0062).

Finding resemblances and dependency as well as meaning of a partial structure and similar meaning as the identical pattern is represented by the disambiguation of “PC” and “personal computer” in Kumai. That is, Kumai explicitly teaches

"means for generating a sentence structure from an input document",

"means for generating a similar structure of patterns having a similar meaning of a partial structure of the sentence structure by performing predetermined conversion operation",

"means for determining the patterns having the similar meaning as the identical pattern and detecting the patterns."

However, Examiner agrees that Kumai fails to teach "the parallel modification being structure modification carrying out the change in connection of the branches so that connecting relationships of the branches in the sentence structure are equal to

each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure". Therefore, Examiner has incorporated Nagao as previously cited to address this limitation not taught by Kumai.

Argument (page 6 ¶ 1):

- "Therefore, the "structural conversion operation" in Akers is not corresponding to the "predetermined conversion operation" in the present invention at all. Akers also neither discloses nor teaches "parallel modification" carrying out the change in connection of the branches in a graph structure as welt as "means for performing non-directional branching" and "means for performing non-ordering of ordering trees" as recited in Claims 1 and 12"

Response to argument:

Examiner disagrees and maintains the use of Akers, wherein Akers builds upon the teachings of Kumai to further effect the conversion of terms by creating trees with various interpretations of not only synonyms but disambiguation of meaning, such that Akers teaches a predetermined conversion operation change in connection of branches in a graph structure, of the partial structure. Further Akers teaches means for performing non-directional branching and means for performing non-ordering of ordering trees and a conversion operation including a change in connection of branches in a graph structure, of the partial structure. Examiner believes that limitations directed to non-directional branching and non-ordering are in fact ambiguous and Examiner

interprets non-directional branchig as analyzing individual words to be analyzed as well as any associated word in a phrase or the phrase itself to derive a semantic meaning rather than solely syntactic, and non-ordering of trees is understood as considering multiple tree interpretations when disambiguating text in view of tree branches with multiple semantic meanings.

Fort instance, Akers teaches well known uses of structural conversions where that each source-language parse tree that involves application of the rule is structurally converted in such a way that the order of the verb and the object is reversed because the verb appears to the right of its object in Japanese. This method is very efficient in that it is easy to find out where the specified conversion applies; it applies exactly at the location where the rule has been used to obtain the source-language parse tree. On the other hand, it can be a weak conversion mechanism in that its domain, as specified above, may be extremely limited, and in that natural language may require conversion rules that straddle over nodes that are not siblings. In template-to-template structural conversion, structural conversion is specified in terms of input/output (I/O) templates or subtrees. If a given input template matches a given structure tree, that portion of the structure tree that is matched by the template is changed as specified by the corresponding output template. This is a very powerful conversion mechanism, but it can be costly in that it can take a long period of time to find out if a given input template matches any portion of a given structure tree (Akers Col 1 lines 34-58).

Akers improves the drawbacks of these well known structural conversion, meaning, and translation techniques. For instance Akers teaches an example where the semantic property of "an arrow" (symbol vs. weapon) is used in evaluating the verb

phrase "like an arrow" in the sentence "They would like an arrow." In contrast, if the syntax of the phrase "an arrow" were changed as in "He shot it with an arrow," the semantic property of "an arrow" is not used in evaluating the verb phrase "shot it with an arrow." (Akers Col 7 lines 9-15).

Further, Akers teaches for example, "cat" and "dog" are more related than "cat" and "pudding", and hence the former pair would be separated by a smaller distance within the tree. "Animal" and "cat" are examples of words that are stored at different levels in the tree, as "animal" is a more abstract term than "cat." (Akers Fig. 9 with Col 10 line 59 to Col 11 line 32). As shown in Fig. 5 below, the structure of a sentence in considered in view of a domain, so that the word “Bank” is interpreted as a side of a river or a financial institution.

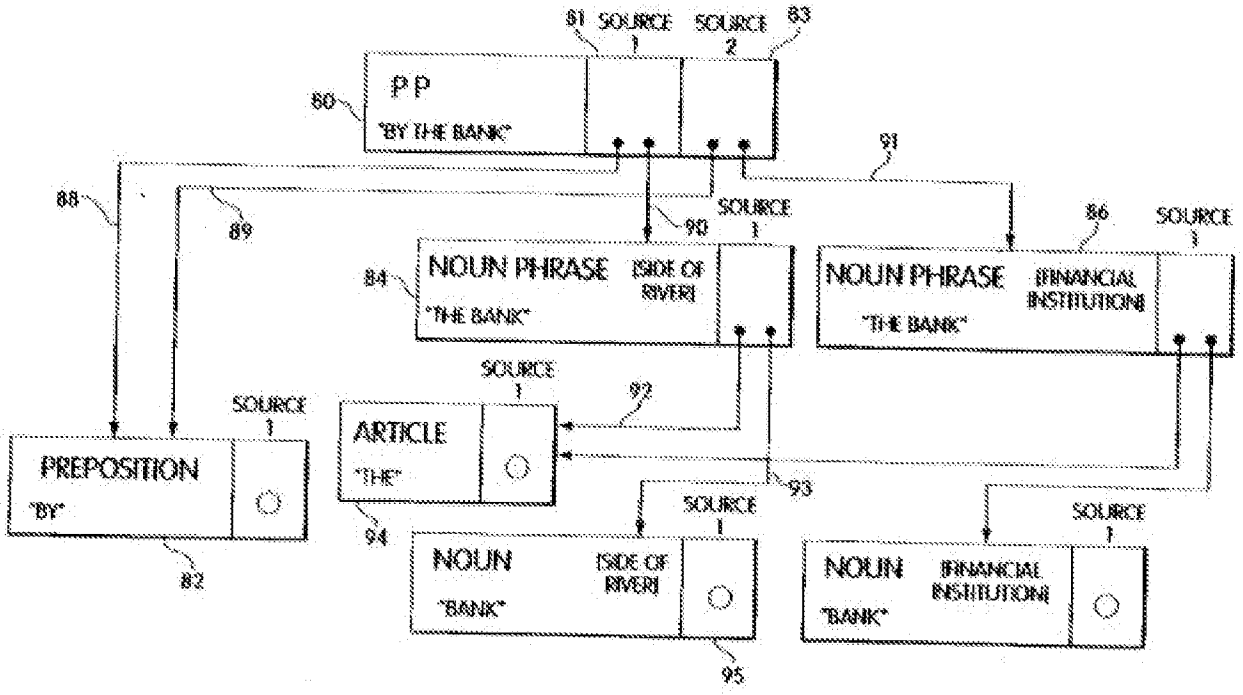

Fig. 9

Furthermore, Akers teaches multiword "idioms" as part of the dictionary, while retaining representations of the individual words of which they are composed. These two forms may ultimately compete against each other to be the best representation. For instance "black sheep" is found in the dictionary with the meaning of a disfavored person. But in some cases the words "black sheep" may refer to a sheep which is black. Because both of the forms are retained, this non-idiomatic usage may still be chosen as the correct translation (Akers Col. 13 lines 1-10).

For instance, Akers teaches a scoring method that applies to all partial sentences f any length, not just full sentences. An important element in the use of a graph is that a subtree is fully scored and analyzed only once, even though it may appear in a great many sentences. For example, in the phrase "Near the bank there is a bank.", the phrase "Near the bank" has at least two meanings, but the best interpretation of that phrase is determined only once. The phrase "there is a bank" similarly has two interpretations, but the best of those two is determined only once. There are therefore four sentence interpretations, but the subphrases are scored just once. Another feature of the graph is that each node is labeled with easily accessible information about the length of that piece of the sentence. This allows the best N interpretations of any substring of the English sentence to be found without reanalyzing the sentence (Akers Col 13 lines 20-40).

Therefore, Examiner believes that Akers teaches means for performing non-directional branching and means for performing non-ordering of ordering trees and a conversion operation including a change in connection of branches in a graph structure, of the partial structure, wherein the teachings of Akers thus far read upon “performing

non-directional branching” and "performing non-ordering of ordering trees”, wherein non-directional branching allows for individual words to be analyzed as well as any associated word in a phrase, and non-ordering of trees allows for multiple trees to be considered when disambiguating text.

However, Examiner agrees that Akers fails to teach "the parallel modification being structure modification carrying out the change in connection of the branches so that connecting relationships of the branches in the sentence structure are equal to each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure”. Therefore, Examiner has incorporated Nagao as previously cited to address this limitation not taught by Akers.

Argument (page 7 ¶ 1):

- “Nagao is an explanatory diagram of the co-occurrence relationship between a path and a sentence (see, col. 4, lines 20-21). This is not an explanatory diagram of "parallel modification" of the present invention. In the present invention, as shown in Fig. 18, "parallel modification" carries out the change in connection of the branches in a graph structure. Accordingly, Nagao neither discloses nor teaches "parallel modification" carrying out the change in connection of the branches, as recited in the claims of the present invention”

Response to argument:

Examiner disagrees and maintains the use of Nagao. What is claimed is “performing parallel modification of the sentence structure, the parallel modification being structure modification carrying out the change in connection of the branches so that connecting relationships of the branches in the sentence structure are equal to each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure”

Examiner believes that Nagao teaches this claim language in any interpretation, wherein Nagao reads upon the core element of the limitation “connecting relationships of the branches in the sentence structure are equal to each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure” as defining parallel modification. Examiner understands this to mean composing another modified structure of a sentence with equal relationship to the original sentence while preserving meaning between all nodes.

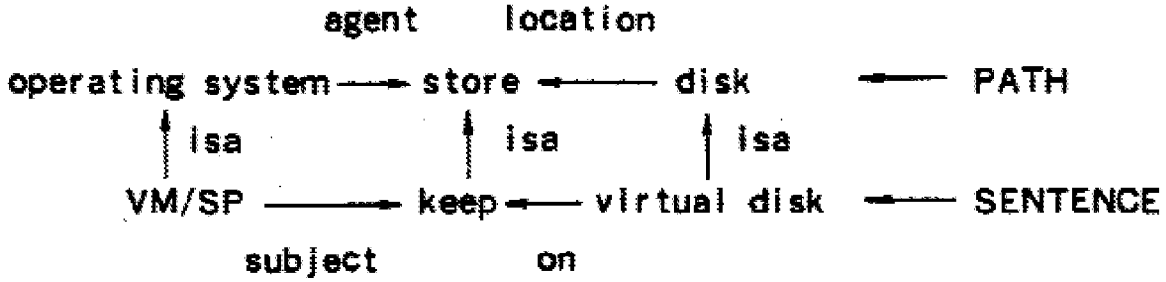

FIG. 21

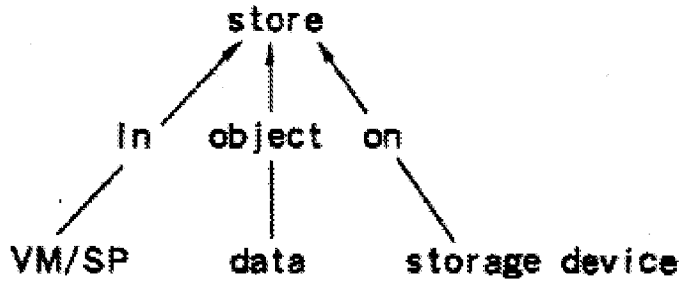

FIG. 22

For instance Nagao teach the resolving of ambiguities as shown above in Fig. 21 and 22, Nagao teaches that if the sentence, "In VM/SP, the data is stored in the storage device.", comes before the above sentence, "VM/SP keeps the information on the virtual disk.", then the dependency structure shown in FIG. 22 is stored as the context data (32) of the knowledge base 30 (the "object" referred to therein is not a semantic case but a grammatical case indicative of an object). If a path is sought between "store" and "disk" of the dependency "store.rarw.disk" appearing in the path of FIG. 16, using

synonym/taxonym relationships and context dependencies of the knowledge base 30, then the path shown in FIG. 23 is found, and it is found that the dependency between "store" and "disk" is defined in the context. Thus the number of dependencies contained in the path and defined in the context is the value of context consistency. In this example, since one dependency is contained in the path and defined in the context, the value of context consistency of the path is 1 (Nago Col 9 lines 21-38).

In Fig. 2, notice how the meaning is preserved when two structures are produced, with one being a parallel modification, such that the context of the remaining sentences encountered are disambiguated, i.e. VM/SP is an operating system, keep is to store, virtual disk is a disk or storage device. Therefore, Nagao improves the teachings of Kumai and Akers to include preserved meaning between nodes in a tree structure when two structures are produced, with one being a parallel modification (i.e. simultaneous), such that the context of the remaining sentences encountered are disambiguated respective of the previous nodes or branches, such that Nagao teaches performing parallel modification of the sentence structure, the parallel modification being structure modification carrying out the change in connection of the branches so that connecting relationships of the branches in the sentence structure are equal to each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1, 2, 4, 5, 7-10, 12, 13, 15, 16, and 19-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kumai US 20040260979 A1 (hereinafter Kumai) in view of Akers et al. US 6278967 B1 (hereinafter Akers) and further in view of Nagao et al. US 5424947 A (hereinafter Nagao).

Re claims 1 and 12, Kumai teaches text mining apparatus comprising:

means for generating a sentence structure from an input document ([0062] & Fig. 5);

the sentence structure representing a dependency among words

means for generating a similar structure of patterns having a similar meaning of a partial structure of the sentence structure by performing predetermined conversion operation ([0062] & Fig. 5),

including at least change in connection of branches in a graph structure, of the partial structure; and

means for determining the patterns having the similar meaning as the identical pattern and detecting the pattern ([0062] & Fig. 5)

means for performing parallel modification of the sentence structure ([0062] & Fig. 5)

the parallel modification being structure modification carrying out the change in connection of the branches so that connecting relationships of the branches in the sentence structure are equal to each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure

wherein the means for generating the similar structure comprises:

means for performing parallel modification of the sentence structure ([0062] & Fig. 5, "content" variations of meaning with respect to the seed sentence);

means for generating a partial structure of the sentence structure ([0062] & Fig. 5 resemblance degree for partial resemblance);

means for replacing a synonym in the sentence structure and/or partial structure by referring to a synonym dictionary ([0062] & Fig. 5); and

the means for generating the similar structure uses the similar structures as an equivalent class ([0008] classification) of the partial structure of the sentence structure ([0062] & Fig. 5)

However, Kumai fails to teach including at least change in connection of branches in a graph structure, of the partial structure.

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic

categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate including at least change in connection of branches in a graph structure, of the partial structure, means for performing non-directional branching of a directional branch of the sentence structure and/or partial structure and means for performing non-ordering of ordering trees of the sentence structure and/or partial structure as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

However, Kumai in view of Akers fails to teach the sentence structure representing a dependency among words

the parallel modification being structure modification carrying out the change in connection of the branches so that connecting relationships of the branches in the sentence structure are equal to each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure

Nagao teaches a natural language analysis apparatus according to the invention can be used for sentence analysis in machine translation systems, question-and-answer

systems using natural languages, and so on, to output the most preferable syntactic tree, in response to an incoming sentence that includes structural ambiguities, by using knowledge on synonym relationships, taxonym relationships, and dependencies among words. Consequently, the system can be used to solve problems that are insoluble by granular-based analysis, such as ambiguities that can only be solved by the use of expert knowledge proper to a specific field or by referring to the contents of a preceding sentence (Nagao Col 3 lines 23-35).

Further, Nagao teaches the connection between branches, wherein the relationship is maintained with a synonymous understanding such as “operating system” and “VM/SP”… “Store” and “keep”, and “disk” and "virtual disk”. The two structures generated are in parallel with each other and dependency maintained (i.e. subject, verb, etc) even when modification of the elements takes place (Nagao Fig. 1 & 21).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai in view of Akers to incorporate the sentence structure representing a dependency among words

the parallel modification being structure modification carrying out the change in connection of the branches so that connecting relationships of the branches in the sentence structure are equal to each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure

as taught by Nagao to allow for a relationship between sentence structure from a text to be maintained with a synonymous understanding, wherein grammatical cases can have the semantic case to maintain consistency also holds, and thus improving the system of Kumai to preserve the dependency structure without ambiguity with respect to

context and synonymous transformation of sentential elements (Nagao Col 3 lines 23-35 with Fig. 1 and 21).

Re claims 2 and 13, Kumai teaches a text mining apparatus according to Claim 1, further comprising:

a storage unit that stores a set of documents ([0062] & Fig. 5) as a text mining object ([0008]); and

an analyzing unit that inputs and analyzes the document of the storage unit and obtains the sentence structure ([0062] & Fig. 5),

wherein the analyzing unit analyzes the document, and generates the sentence structure ([0062] & Fig. 5) containing a clause having a node and indicating at least a dependency as a directional branch from the node on a modifier to the node on a modifiee.

However, Kumai fails to teach a clause having a node and indicating at least a dependency as a directional branch from the node on a modifier to the node on a modifiee.

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible

syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate a clause having a node and indicating at least a dependency as a directional branch from the node on a modifier to the node on a modifiee as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Re claims 4, 15, and 23, Kumai teaches a text mining apparatus comprising:

a storage unit that stores a set of documents ([0062] & Fig. 5) as a text mining object ([0008]);

an analyzing unit that reads and analyzes the document from the storage unit and obtains the sentence structure ([0062] & Fig. 5);

a similar-structure generating unit that performs predetermined modification operation ([0062] & Fig. 5),

including at least change in connection of branches in a graph structure,

of the partial structure of the sentence structure obtained by the analysis of the analyzing unit, and generates a similar structure of patterns having a similar meaning ([0062] & Fig. 5); and

a pattern detecting unit that uses the similar structure generated by the similar-structure generating unit as an equivalent class of the partial structure on the generation source, and detects the pattern ([0062] & Fig. 5)

wherein the means for generating the similar structure comprises:

means for performing parallel modification of the sentence structure ([0062] & Fig. 5, "content" variations of meaning with respect to the seed sentence);

means for generating a partial structure of the sentence structure ([0062] & Fig. 5 resemblance degree for partial resemblance);

means for replacing a synonym in the sentence structure and/or partial structure by referring to a synonym dictionary ([0062] & Fig. 5); and

the means for generating the similar structure uses the similar structures as an equivalent class ([0008] classification) of the partial structure of the sentence structure ([0062] & Fig. 5)

However, Kumai fails to teach including at least change in connection of branches in a graph structure, of the partial structure.

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible

syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate including at least change in connection of branches in a graph structure, of the partial structure as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

However, Kumai fails to teach means for performing non-directional branching of a directional branch of the sentence structure and/or partial structure and

means for performing non-ordering of ordering trees of the sentence structure and/or partial structure

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible

interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have also been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate means for performing non-directional branching of a directional branch of the sentence structure

and/or partial structure and means for performing non-ordering of ordering trees of the sentence structure and/or partial structure as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

However, Kumai in view of Akers fails to teach the sentence structure representing a dependency among words

the parallel modification being structure modification carrying out the change in connection of the branches so that connecting relationships of the branches in the sentence structure are equal to each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure

Nagao teaches a natural language analysis apparatus according to the invention can be used for sentence analysis in machine translation systems, question-and-answer systems using natural languages, and so on, to output the most preferable syntactic tree, in response to an incoming sentence that includes structural ambiguities, by using knowledge on synonym relationships, taxonym relationships, and dependencies among words. Consequently, the system can be used to solve problems that are insoluble by granular-based analysis, such as ambiguities that can only be solved by the use of

expert knowledge proper to a specific field or by referring to the contents of a preceding sentence (Nagao Col 3 lines 23-35).

Further, Nagao teaches the connection between branches, wherein the relationship is maintained with a synonymous understanding such as “operating system” and “VM/SP”… “Store” and “keep”, and “disk” and "virtual disk”. The two structures generated are in parallel with each other and dependency maintained (i.e. subject, verb, etc) even when modification of the elements takes place (Nagao Fig. 1 & 21).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai in view of Akers to incorporate the sentence structure representing a dependency among words

the parallel modification being structure modification carrying out the change in connection of the branches so that connecting relationships of the branches in the sentence structure are equal to each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure

as taught by Nagao to allow for a relationship between sentence structure from a text to be maintained with a synonymous understanding, wherein grammatical cases can have the semantic case to maintain consistency also holds, and thus improving the system of Kumai to preserve the dependency structure without ambiguity with respect to context and synonymous transformation of sentential elements (Nagao Col 3 lines 23-35 with Fig. 1 and 21).

Re claims 5 and 16, Kumai teaches a text mining apparatus according to Claim 4, wherein the pattern detecting unit uses the similar structure as the equivalent class of the partial structure on the generation source, and detects the pattern ([0062] & Fig. 5).

However, Kumai fails to teach means for performing non-ordering of ordering trees in the sentence structure and/or partial structure

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate means for performing non-ordering of ordering trees in the sentence structure and/or partial structure as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Re claims 7 and 18, Kumai teaches a text mining apparatus ([0008]) according to Claim 4, further comprising:

However, means for adjusting the operation so that a user determines how similar patterns are identical and detecting the pattern

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring

module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate adjusting the operation so that a user determines how similar patterns are identical and detecting the pattern as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Re claims 8, 19, 24, and 25, Kumai teaches a text mining apparatus comprising:

a storage unit that stores a set of documents ([0062] & Fig. 5) as a text mining object ([0008]);

an analyzing unit that reads and analyzes the document from the storage unit and obtains a sentence structure ([0062] & Fig. 5);

whether or not the structures are identical one every type of differences between the sentence structures ([0062] & Fig. 5);

whether or not the structures are identical ones every type of differences between attribute values ([0062] & Fig. 5);

a similar-structure generating unit that performs predetermined conversion operation of a partial structure of the sentence structure obtained by the analyzing unit in accordance with the first determination item generated by the similar-structure generation adjustment unit and generates similar structures having a similar meaning of the partial structure ([0062] & Fig. 5); and

a similar-pattern detecting unit that uses the similar structure generated by the similar-structure generating unit as an equivalent class of the partial structure on the generation source and detects the frequent pattern ([0060] & Fig. 3 frequency) by ignoring the difference between the attribute values in accordance with the second determination item of the similar-structure determination adjustment unit ([0062] & Fig. 5).

wherein the similar- structure generating unit comprises:

means for performing parallel modification of the sentence structure when the first determination item determines the parallel modification ([0062] & Fig. 5);

means for generating a partial structure of the sentence structure ([0062] & Fig. 5 resemblance degree for partial resemblance);

means for replacing a synonym in the sentence structure and/or partial structure by referring to a synonym dictionary when the first determination item includes replacement of the synonym ([0062] & Fig. 5); and

the similar-structure generating unit generates a similar structure of the sentence structure and sets the similar structure as the equivalent class ([0062] & Fig. 5)

However, Kumai fails to teach

means for performing non-directional branching of a directional branch of the sentence structure and/or partial structure when the first determination item determines the non-directional branching of the directional branch;

means for performing non-ordering of ordering trees of the sentence structure and/or partial structure when the first determination item determines the non-ordering of the ordering trees, and wherein;

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text.

The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate means for performing

non-directional branching of a directional branch of the sentence structure and/or partial structure when the first determination item determines the non-directional branching of the directional branch and means for performing non-ordering of ordering trees of the sentence structure and/or partial structure when the first determination item determines the non-ordering of the ordering trees, and wherein as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

However, Kumai fails to teach a similar-structure generation adjustment unit that generates a first determination item for determining, from a user input,

a similar-structure determination adjustment unit that generates a second determination item for determining, from a user input,

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes

nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have also been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate a similar-structure generation adjustment unit that generates a first determination item for determining, from a user input and a similar-structure determination adjustment unit that generates a second determination item for determining, from a user input as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

However, Kumai in view of Akers fails to teach the sentence structure representing a dependency among words

the parallel modification being structure modification carrying out the change in connection of the branches so that connecting relationships of the branches in the sentence structure are equal to each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure

Nagao teaches a natural language analysis apparatus according to the invention can be used for sentence analysis in machine translation systems, question-and-answer systems using natural languages, and so on, to output the most preferable syntactic tree, in response to an incoming sentence that includes structural ambiguities, by using knowledge on synonym relationships, taxonym relationships, and dependencies among

words. Consequently, the system can be used to solve problems that are insoluble by granular-based analysis, such as ambiguities that can only be solved by the use of expert knowledge proper to a specific field or by referring to the contents of a preceding sentence (Nagao Col 3 lines 23-35).

Further, Nagao teaches the connection between branches, wherein the relationship is maintained with a synonymous understanding such as “operating system” and “VM/SP”… “Store” and “keep”, and “disk” and "virtual disk”. The two structures generated are in parallel with each other and dependency maintained (i.e. subject, verb, etc) even when modification of the elements takes place (Nagao Fig. 1 & 21).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai in view of Akers to incorporate the sentence structure representing a dependency among words

the parallel modification being structure modification carrying out the change in connection of the branches so that connecting relationships of the branches in the sentence structure are equal to each other for all of nodes corresponding to the words put in a parallel relationship in the sentence structure

as taught by Nagao to allow for a relationship between sentence structure from a text to be maintained with a synonymous understanding, wherein grammatical cases can have the semantic case to maintain consistency also holds, and thus improving the system of Kumai to preserve the dependency structure without ambiguity with respect to context and synonymous transformation of sentential elements (Nagao Col 3 lines 23-35 with Fig. 1 and 21).

Re claims 9 and 20, Kumai teaches a text mining apparatus ([0008]) according to Claim 8, wherein the analyzing unit analyzes the document ([0062] & Fig. 5), and generates the sentence structure containing a clause having a node and indicating at least a dependency as a directional branch from the node on a modifier to the node on a modifiee determination,

an the attribute value includes the surface case and/or the information about the attached word, added to the sentence structure ([0062] & Fig. 5).

However, Kumai fails to teach generating the sentence structure containing a clause having a node and indicating at least a dependency as a directional branch from the node on a modifier to the node on a modifiee determination.

Akers teaches a translation engine includes a preparer, a parser, a graph maker, an evaluator, a graph scorer, a parse extractor, and a structural converter. The preparer examines the input text and resolves any ambiguities in input sentence boundaries. The preparer then creates and displays the input text in a parse chart seeded with dictionary entries. The parser parses the chart to obtain possible syntactic categories for the input text. The graph maker produces a graph of the possible syntactic interpretations of the input text based on the parse chart. The graph includes nodes and subnodes which are associated with possible interpretations of the input text. The evaluator, which comprises a series of experts, evaluates the graph of the possible interpretations and adds expert weights to the nodes and subnodes of the graph. The graph scorer uses the expert weights to score the subnodes, and the graph scorer then associates the N best scores with each node. The parse extractor assigns a parse tree

structure to the preferred interpretation as determined by the graph scorer. The structural converter performs a structural conversion operation on the parse tree structure to obtain a translation in the target language (Akers Col. 4 lines 41-67).

Further, Akers teaches that the order in which nodes are visited and scored is a standard depth-first graph-walking algorithm. In this algorithm, nodes that have been scored are marked and are not scored again. During the scoring process, the scoring module evaluates dictionary entry nodes before evaluating any of the higher unit nodes. Each dictionary entry gives rise to a single score. For a unary rule, each of the k scores of the lower export is added to the expert values that apply to the unary rule, and the resulting vector of k scores is associated with the parent export. For a binary rule, assume that the left child has g scores and the right child has h scores. Then a total of g times h scores are computed by adding each of the left child's scores to each of the right child's scores, and in addition, adding the expert values that apply to the binary rule. When g times h exceeds N, only the N best scores are kept with the parent node (Akers Col. 8 lines 6-12).

Furthermore, Akers teaches user interaction which can change the conversion of a sentence and its corresponding tree structure generation (Akers Col. 9 lines 1-45).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Kumai to incorporate generating the sentence structure containing a clause having a node and indicating at least a dependency as a directional branch from the node on a modifier to the node on a modifiee determination, as taught by Akers to allow for a user interface (Akers Col. 8 lines 6-12) having the ability to generate a tree structure which compares various

sentential outputs depending on scores within the nodes of a tree structure (Akers Col. 4 lines 41-67), wherein the order of words in a sentence may not reflect the best meaning and can be varied to produce the best scoring tree with the same overall structure but with a better meaning (Akers Col. 8 lines 6-12) based on user preferences.

Re claims 10 and 21, Kumai teaches a text mining apparatus according to Claim 8, wherein the similar-pattern detecting unit detects a frequent similar pattern ([0060] and [0062] & Fig. 3 and 5).

Conclusion

4. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MICHAEL C. COLUCCI whose telephone number is (571)270-1847. The examiner can normally be reached on 8:30 am - 5:00 pm , Monday - Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Richemond Dorvil can be reached on (571)-272-7602. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Michael C Colucci/
Examiner, Art Unit 2626
Patent Examiner
AU 2626
(571)-270-1847
Examiner FAX: (571)-270-2847

Michael.Colucci@uspto.gov